In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
# Load the dataset
file_path = r"C:\Users\mogut\Downloads\insurance.csv"
df = pd.read_csv(file_path)

# Display the first few rows of the dataset
df.head()
```

Out[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [3]:

```
# Check for missing values
print(df.isnull().sum())

# Convert categorical variables into dummy/indicator variables
df = pd.get_dummies(df, drop_first=True)

# Display the first few rows after encoding
df.head()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Out[3]:

	age	bmi	children	charges	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.900	0	16884.92400	False	True	False	False	True
1	18	33.770	1	1725.55230	True	False	False	True	False
2	28	33.000	3	4449.46200	True	False	False	True	False
3	33	22.705	0	21984.47061	True	False	True	False	False
4	32	28.880	0	3866.85520	True	False	True	False	False

In [5]:

```
# Define features (X) and target (y)
X = df.drop("charges", axis=1)  # Assuming 'charges' is the target variable
y = df['charges']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [6]:

```
# Initialize the linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)
```

Out[6]:

```
▾ LinearRegression

LinearRegression()
```

In [7]:

```
# Make predictions
y_pred = model.predict(X_test)
```

```python
# Calculate Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

```
Mean Squared Error: 33596915.851361446
```